What is claimed is:

1. A method of processing a transaction, the method comprising:

processing a transaction workload in a primary process pair on a first node in a cluster of nodes, the processing using a database stored on at least one stable storage volume and a log stored on another stable storage volume, the at least one stable storage volume and the log storage volume forming a log storage group;

while processing the transaction workload, performing checkpointing operations via the network from the primary process pair to a backup process pair, the backup process pair operating on a second node in the cluster of nodes;

detecting a failure making the first node inoperable or inaccessible; and

after detecting the failure, engaging the backup process pair to take over the transaction processing workload of the primary process pair, the backup process pair being configured to operate with the log storage group used by the primary process pair on the failed node.

2. The method of claim 1, wherein performing checkpointing operations includes communicating checkpointing information by the primary process pair to the backup process pair.

3. The method of claim 1, each node in said cluster including at least one server, wherein one of said at least one servers hosts the backup process pair, the hosting server having a cache for holding transaction data; and

wherein the cache is loaded with transaction data derived from the log storage volume of the log storage group to prepare the cache for use by the backup process pair when the backup pair takes over the work of the primary process pair.

4. A method of transferring a transaction processing workload, the method comprising:

establishing a plurality of primary process pairs on a first node in a cluster of nodes, each node including one or more servers and being associated with a plurality of stable storage volumes, the plurality of primary process pairs including a plurality of first primary processes and a plurality of first backup processes;

grouping the plurality of stable storage volumes of the first node into a number of groups and assigning a separate audit trail to each group of stable storage volumes, each of the separate audit trails and their associated group of stable storage volumes forming a log storage group, the separate audit trail for each log storage group recording stable storage updates for the stable storage volumes in the log storage group;

establishing a plurality of backup process pairs on a second node in the cluster, the plurality of backup process pairs including a plurality of second primary processes and a plurality of second backup processes;

if the node on which the plurality of backup process pairs is established is in communication with the node hosting the plurality of primary process pairs, performing checkpointing operations via the network from the plurality of primary process pairs to the plurality of backup process pairs;

performing checkpointing operations on the node hosting the plurality of primary process pairs;

detecting a failure that makes the first node inoperable or inaccessible;

after detecting the failure, engaging the plurality of backup process pairs as a new plurality of primary process pairs to perform the transaction processing workload of the plurality of primary process pairs in the failed node, the plurality of backup process pairs being configured to operate with a log storage group associated with the plurality of primary process pairs on the failed node.

5. A method of transferring a transaction processing workload as recited in claim 4, wherein prior to engaging the backup process pair, if the plurality of backup process pairs is not present, the method further comprises creating a plurality of backup process pairs on the second node.

6. A method of transferring a transaction processing workload as recited in claim 4,

wherein the node that hosts the plurality of new primary process pairs has a cache for holding transaction data; and

wherein the cache is loaded with transaction data derived from the audit trail to prepare the cache for operation with the plurality of new primary process pairs.

7. A method of transferring a transaction processing workload as recited in claim 4, wherein performing checkpointing operations on the node hosting the plurality of primary process pairs includes:

checkpointing transaction updates from the plurality of first primary processes to the plurality of first backup processes;

writing transaction updates and a communications flag to the audit trail, the communications flag indicating whether the node hosting the plurality of primary process pairs is in communication with the node on which the plurality of backup process pairs is established; and

writing transaction updates to said plurality of stable storage volumes.

8. A method of transferring a transaction processing workload as recited in claim 4, wherein performing checkpointing operations from the plurality of primary process pairs to the plurality of backup process pairs includes:

checkpointing transaction updates from the plurality of primary process pairs to the plurality of backup process pairs;

writing transaction updates and a communications flag to the audit trail, the communications flag indicating whether the node hosting the plurality of primary process pairs is in communication with the node on which the plurality of backup process pairs is established; and

writing transaction updates to said plurality of stable storage volumes.

9. A method of transferring a transaction processing workload as recited in claim 4,

wherein the node hosting the plurality of primary process pairs includes a primary audit process pair that performs logging operations for the plurality of primary process pairs;

wherein the node on which the plurality of backup process pairs is established has a backup audit process pair that performs logging operations for the plurality of backup process pairs after takeover; and

further comprising checkpointing audit trail information from the primary audit process pair to the backup process pair.

10. A method of transferring a transaction processing workload as recited in claim 4,

wherein the node hosting the plurality of primary process pairs includes a primary audit process pair that maintains a list of sequential updates in an audit trail for the plurality of primary process pairs;

wherein the node on which the plurality of backup process pairs is established has a backup audit process pair that maintains a list of sequential updates in an audit trail for the plurality of backup process pairs;

further comprising writing each update in the audit trail to a volume in the log storage group if the node on which the plurality of backup process pairs is established is out of communication with the node hosting the plurality of primary process pairs.

11. A method of transferring a transaction processing workload as recited in claim 4,

wherein the node that hosts the plurality of new primary process pairs has a cache for holding transaction data;

wherein, if, based on the communication flag in the audit trail, the node on which the plurality of backup process pairs is established was out of communication with the node hosting the plurality of primary process pairs prior to the failure, said engaging the plurality of backup process pairs includes performing a recovery process based on data in the audit trail of the log storage group to prepare the cache for operation with the plurality of new primary process pairs.

12. A method of transferring a transaction processing workload as recited in claim 11,

wherein the node that hosts the plurality of new primary process pairs has a cache for holding transaction data; and

wherein performing a recovery procedure includes performing redo and undo operations based on updates in the audit trail of the log storage group to prepare the cache.

13. A method of transferring a transaction processing workload as recited in claim 12,

wherein the node that hosts the plurality of new primary process pairs has a cache for holding transaction data and the cache is operated according to a "steal, no force" policy; and

wherein the audit trail includes at least two checkpoints and the undo and redo operations, performed based on the updates in the audit trail, stop at the penultimate checkpoint.

14. A method of transferring a transaction processing workload as recited in claim 4,

wherein the node that hosts the plurality of new primary process pairs has a cache for holding transaction data; and

wherein, if the plurality of backup process pairs is not present, said engaging the plurality of backup process pairs includes performing a recovery process based on data in the audit trail of the log storage group to prepare the cache for operation with the plurality of new primary process pairs.

15. A method of transferring a transaction processing workload as recited in claim 14, wherein performing a recovery procedure includes performing redo and undo operations based on updates in the audit trail of the log storage group.

16. A method of transferring a transaction processing workload as recited in claim 15,

wherein the node that hosts the plurality of new primary process pairs has a cache for holding transaction data and the cache is operated according to a "steal, no force" policy; and

wherein the redo and undo operations, performed based on the updates in the audit trail in the log storage group, stop at the beginning of the audit trail.

17. A transaction processing apparatus, comprising:

a plurality of stable database storage volumes, one or more of the storage volumes being organized into a group;

a plurality of stable transaction log storage volumes, each log storage volume being associated with one of said stable storage volume groups to form a log storage group; and

a plurality of connected computing nodes, at least one node having a plurality of primary process pairs for performing work on behalf of a transaction by accessing the log storage group, wherein any of the stable storage volumes and log storage volumes are accessible by any of said computing nodes, and at least one other node having a plurality of backup process pairs for taking over the work of said plurality of primary process pairs by accessing the log storage group used by said plurality of primary process pairs, if said plurality of primary process pairs are non-operational.

18. A transaction processing apparatus as recited in claim 17, wherein the plurality of primary process pairs communicate checkpointing information to the plurality of backup process pairs.

19. A transaction processing apparatus as recited in claim 17,

wherein the computing node with the plurality of backup process pairs has a cache for holding transaction data; and

wherein the cache is loaded with transaction data derived from the log storage volume of the log storage group to prepare the cache for use by the plurality of backup process pairs when the plurality of backup pairs takes over the work of the plurality of primary process pairs.

20. A transaction processing apparatus as recited in claim 17,

wherein the computing node with the plurality of primary process pairs and the computing node with the plurality of backup process pairs each have a cache for holding transaction data, the cache in the computing node with the plurality of backup process pairs being maintained with substantially the same information as the cache in the computing node with the plurality of primary process pairs; and

wherein the cache in the computing node of the plurality of backup process pairs is used by the plurality of backup process pairs when the plurality of backup pairs takes over the work of the plurality of primary process pairs.

21. A fault-tolerant cluster of computing nodes, comprising:

means for stably storing database information, said database storage means including a plurality of storage units organized into a group;

means for stably storing log information, said log storage means including a unit associated with at least one storage unit group;

means for computing in a first node including a primary process pair for performing work on behalf of a transaction by accessing said unit of said associated log storage means and said group of units of said data storage means;

means for computing in a second node, including a backup process pair for taking over the work of said primary process pair by accessing said associated unit of said log storage means and said group of units of said data storage means; and

means for interconnecting first and second nodes, database storage means and log storage means such that any unit of the database storage means and any unit of the log storage means is accessible by the first and second computing means.

22. A fault-tolerant cluster of computing nodes as recited in claim 21,

wherein first computing means and second computing means each include volatile storage means for storing database information, volatile storage means in second computing means having substantially the same information as the volatile storage means in first computing means; and

wherein said backup process pair uses the volatile storage means in the second computing means in taking over the work of said primary process pair.